Published Date Extraction System a Semi-Supervised Approach of Extraction

[1]Nitin Kumar, [2]Abhishek Pradhan

Abstract*— *The need to extract a meaningful or relevant dates like published date from an unstructured document is a very vital cog in the wheel of information extraction and data mining field. The current approaches usage DOM (Document Object Model) manipulation for an HTML document or regex expression and rules from metadata which are not so accurate for different types of publication. The recent work in this area mainly focused on web pages and HTML pages with some good accuracy. Our approach took a leaf from those works for HTML, and along with that it extensively covers PDF document, Blog articles, and Websites. It supports several types of documents like News Articles, Patents, Scientific Articles/Journal in PDF format, Blogs, Websites and more. It also has the capabilities to learn over the period and feed the learnings back to the system as trained model. Our algorithm comprises of both supervised and unsupervised steps, and it uses natural language processing techniques.

Keywords— *CRF modelling for Segment extraction, data mining, information extraction, published date*

I. INTRODUCTION

The need to group search results based on a published date of a web document (HTML or PDF) has increased many folds due to aptness or relevancy of the result based on date. If we analyze research areas in any domain, we can easily understand why researcher or practitioner want to read only current journals and papers or blogs. Technology landscape changes rapidly and within a year some innovation, and findings turned out to be obsolete. Hence it is imperative for any content aggregator to sort documents based on published date before providing the result to their user. The researcher always needs the latest update in their research area hence they are more interested in new journals and research papers on their topic. Similarly, if there are any news articles, everyone wants to read the latest news but in the case of blogs both previous and current topics are required for complete details regarding a subject. We tried to extract published date based on document type because the meaning changes based on document type. For example, the published date for journals, research and news articles are considered as last updated date at the same time for the book it is first updated date on the web. Similarly, for websites and blogs, the first indexed time is considered as published date. We have examined these differences in our approach to solving the problem. The extraction of Published date is harder for a PDF

document. The current methods consider mostly metadata and relying only on metadata is not the correct choice because several web documents don't contain the published date attribute or it may list the wrong date. We are analyzing the PDF content in details by dividing it into different segments and choosing the right section for published date. This divide and rule policy for PDF helps us to reduce the processing time as we don't process all parts of PDF for published date extraction. The another approach for PDF document is to extract relevant dates manually and store it for processing. It gives a good result but it is time-consuming, and hence it is not a scalable solution. In the case of HTML document, the approach explained in [1] is to identify features related to dates from DOM element of a document and identify published dates based on CRF trained model. It yielded good results, but it has its shortcomings. We found several instances where published date was not available in HTML elements or several dates were available as a candidate for published date. We also encountered several cases where the structure or DOM element of an HTML varies based on publication to publication and hence it is not easy to prepare error-free feature set for CRF based modeling. All these concerns for HTML document addressed in [2]. It primarily focused on link-based methods: it systematized and generalized several simple date propagation methods previously proposed in the literature, and it also suggested a more sophisticated likelihood optimization based method also it proposed the data-driven parameter selection approach. In a nutshell, the two prominent problems i.e. First, not all web pages contain the publication dates in their texts and Second, it's hard to distinguish the publication date among all the dates found in the page's text is resolved by [2]. We also took the similar path for the HTML but for PDF documents like scientific journals, research paper, books, Website and Blogs we differ entirely from previous approaches and the complete details of our algorithm is mentioned in section II. The preprocessing stage of our algorithm emulate few logical steps like classification and

segmentation of a document as listed in [3] but our algorithm clearly differs in the implementation of those steps. In the case of PDF document, we use segmentation technique as mentioned in [4],[5] to extract segments from a PDF document as part of preprocessing steps of our algorithm. As per our testing, Grobid provides an excellent accuracy of 82% for segmentations. The accuracy of segmentation using Grobid validated in [11]. We also used Grobid to extract metadata of a PDF document as the accuracy provided by Grobid is better than others as compared in [6]. Metadata extraction is also part of our preprocessing steps. In this paper, we have stated the need for different type of approaches for a different type of documents i.e. it is different for news articles[9] and books and general HTML. We have also provided a unique and extensive approach for PDF document along with the extraction of published date for blogs, websites, books which were not available earlier. Few other methods as mentioned in [8] depends on regular expression for extracting published date which is not an accurate and scalable solution. Our approach is scalable and learnable. We have provided the details of our comparison and performance matrix in comparison section V. Unfortunately, we don't have any existing system to compare with which are mentioned in references or existing work list hence we choose existing commercial system to compare our accuracy. We compared with IBM Alchemy APIs for published date. Currently, Alchemy's API is only meant for HTML documents hence for PDF document we have evaluated our accuracy independently. Our approach is mostly a mixture of the supervised method used for segment extraction and classification of a document combined with an unsupervised method of published date extraction. For segmentation, we used both supervised approach for scientific articles, journal and patents [5],[4] and unsupervised method based on OCR(Optical character recognition) methodology[12] for other random PDF articles to extract segments. It can also learn over the period and create a model automatically for further processing. This paper divides into multiple sections. The first and foremost section provides complete details of all the four building blocks used in this article followed by simple architecture flow diagram. It follows by an exhaustive comparison with Alchemy's API developed by Alchemy an IBM company. In comparison section, we have also provided accuracy of our measure for PDF document. In conclusion section, we summarize our findings. In the last section, we proposed the future implementation of other relevant dates apart from published date.

II. LAYERED ARCHITECTURE

We devised a four-layered approach to solving this problem. The first layer is Extractor Layer based on CRF (Conditional Random Field) model to identify sections from an unstructured document [4],[5]. The second layer is a classification layer which relies on Reuters and Wikipedia-based trained model as described in [13],[14]. We have not described document classification technique in details as it is beyond the scope of this paper. These methods are used to classify documents as News, Book, Blogs, Articles and General documents. The third critical layer is the Processing Layer which does most of the heuristic task like cleaning text, annotate date, convert a document into trigram model and annotate POS as well. This layer also has the capability to identify text like "holiday", "today"," last year" as date tag. We used this information for trigram modeling, parsing and to provide these details to the fourth layer i.e. Rule Engine Layer for identifying important dates. The multilayer architecture designed in such a way that input translates into a knowledgegraph before published date identification process triggers. It helps to exclude noise and prepares an exclusion list. It is imperative that we reduce the candidate set which satisfies the required features without reducing or removing the required information to extract any relevant information from any unstructured text. We applied exclusion strategy (filter at each step to narrow down the candidate list) rather than inclusion policy (extract all dates and add into candidate published date) to arrive at the required result. The fig1 provides a complete detail of all the building blocks of our algorithm.

A. First Layer: Extractor

This layer is completely supervised layer used to identify URL, title, metadata, body text (primary content) and bibliography sections from any unsupervised text. We trained our model using CRF++ and obtained a CRF based model to identify sections from scientific articles. The F1 score for this model is more than 0.9, and hence we got distinct sections with good accuracy for further processing. This layer is critical as it gives only the required text to process further rather than processing complete text. We also identify table and images using OCR (Object character recognition) based APIs and exclude the text available in tables. In our testing, we found that sometimes table contains data which qualifies as date and hence sometimes it affects the right output. It is always better to add as many dates as possible in exclusion list for better accuracy. We also add all the dates available in bibliographic section in exclusion list with an assumption that all referred article and the current article cannot have the same date. We also get inference from reference section that the published date for the current article should be greater than the published date of all referenced article mentioned in the reference section. The first layer strengthens our exclusion

strategy, and hence it helps in obtaining better accuracy from our algorithm.

B. Second Layer: Classifier

As we support multiple types of documents like a news article, book, blogs, scientific journals, patent, and others, it is crucial to understand different segmentation style of each of these. We use Reuters corpus and Wikipedia corpus for classification. It is a simple classification mechanism where we require mostly news, blogs, and books. We used maximum entropy modeling technique provided by OpenNLP to train our model for classification.

C. Third Layer: Processing

The processing layer does several activities like removing stop words from body content, de-hyphenation of all the hyphenated text. It also annotates POS (Parts of speech)and tagged all date instances in a text. We used Stanford NLP for POS and date tagging. All sentences convert into trigram model. We filter the trigram where the date annotated, and a remaining trigram discarded. Hence all the body content is converted into annotated trigram model.

The date extraction from metadata performed in this layer. The date extracted from meta based on various tags of meta as mentioned in the blog [8].

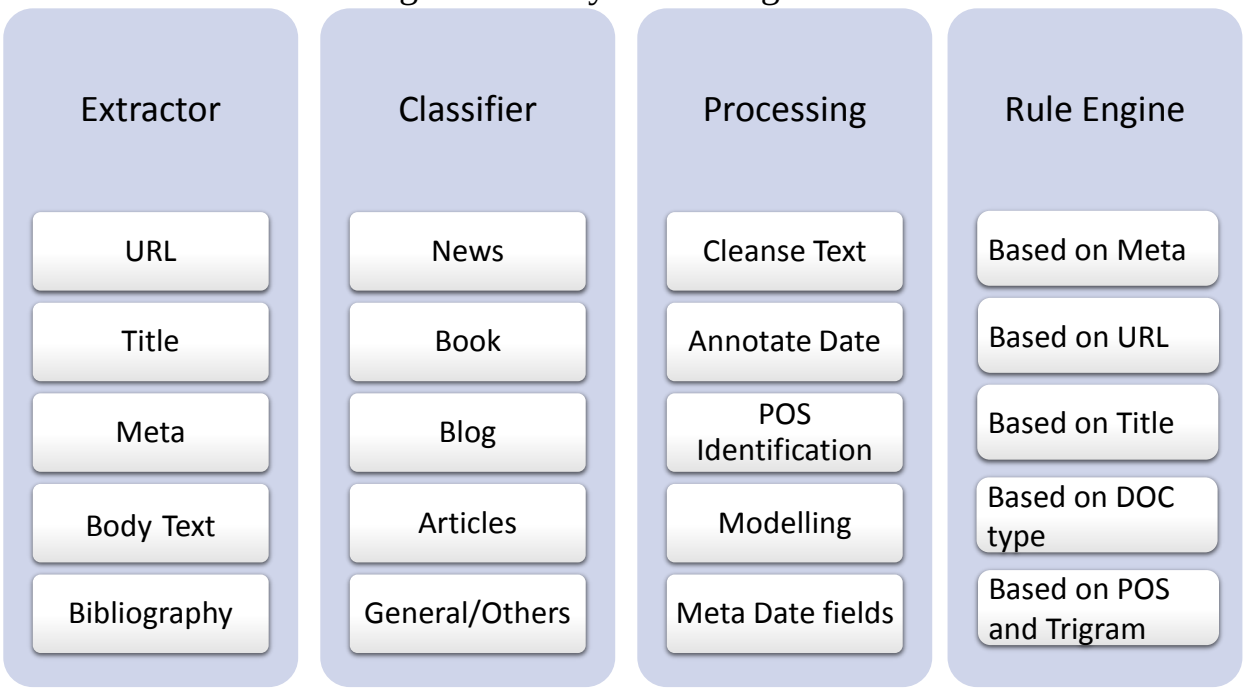
D. Fourth Layer: Rule Engine

The rule engine is the last layer which provides a way to process data derive from other three layers sequentially. This layer helps to handle different document type as per their associated rules. The rules are straightforward and efficient in analyzing published date. The rules are for URL, title, meta, sections and POS based features in trigram. These rules combined with document type rules.

Rule 1:If the document is news [9] type, then we only analyze dates from URL, meta, and title in a sequential manner. If we don't get published date, we analyze content but only the top quadrant of the first page. The idea is to analyze dates mentioned after title but before the main content start. We identify quadrant based on OCR methodology in the case of pdf, and if the news articles are in the form of HTML, then this information are extracted using DOM [1] element parsing.

Rule 2:If the document is book type, then the process changes slightly. We only consider URL, title, meta and extract only the content before TOC (Table of Content) as body content.

Fig 1: Four Layer Building Blocks



then we search crossref.org based on the title and author extracted from the document. If it finds the document based on the title and author in Crossref it means the book comprised of journals and articles published by the author. We choose published date of those articles as published date as mentioned in Crossref.

Rule 4:All the dates mentioned in reference section cannot be a published date and published date should be after/greater than all these dates.

Rule 5:If the document is BLOG type, then we identify the end of each blog mentioned on a BLOG site. It is equivalent to section identification from pdf document. We consider the last date in a BLOG sites as Published date. Section identification is important as all BLOG sites contains several adds and other non-relevant content. Hence it is required to cleanse data before considering the last date as published date.

Rule 6:We always check published dates extracted from multiple sources and compare it before we mark it as a result. As an example, if published date identified from URL and meta then we examined it before marking it as a result. If date matches from two sources, it increases the probability of getting accurate results.

Rule 7:In the case of Scientific articles or another type of documents, if we are unable to identify dates based on all the above rules, we analyze trigram models of text annotated with POS and dates. It only selects those trigrams where the date marked as an entity based on POS sequence and other features published date identified. Details are mentioned in Learnability section III.

Rule 8: At last, we also check the archive date of any URL i.e. when the document site was first indexed or made available on the internet. It mainly helps if a site is a website.

Rule 9: Learnability is an important aspect of our algorithm. As and when trigram of POS sequence identified and stored inside a file then based on threshold value it considered for training and feature selection.

III. LEARNABILITY

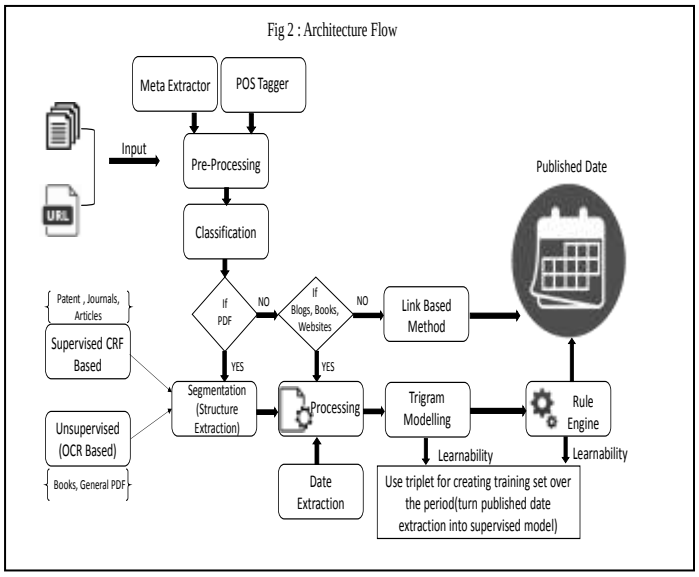
A. *Trigram Feature Extraction*

The training set was created using a set of trigrams [7] having dates. These trigrams manually annotated as positive or negative outcomes. The positive results considered as Candidate Published Date (CPD), and adverse outcomes considered as non-CPD. Multiple features were identified and selected for training classifier using Logistic Regression (LR), a discriminative model of Machine Learning that uses an independent variable (x) to predict the dependent variable (y) and Stochastic Gradient Descent (SGD) for parameter vector (theta) optimization.

B. *Feature Selection and Training*

Our experiment shows trigram frequency feature results in significantly high precision and recall. Trigrams converted to POS tags, and Date Entity tag helped to identify dominant feature over the text. Trigrams are having frequency more than n times only selected for the training set to reduce excess noise from the training set. LR (Logistic Regression) is implemented in Java using Mahout's Online Logistic Regression with several classes and features. We ran the training on Ubuntu 16.04 based PC having 8 GB Ram using Intel Core i3 processor.

Fig 2 : Architecture Flow



The experiment shows, 90 percent of times with document having published date in content body, the published date was identified among top 3 annotated CPD and 70% of times the published date was identified correctly with the highest probability.

IV. DATA FLOW

We combine all the building blocks into a logical binding to arrive at the optimum result. These bindings governed by the rule engine.

Our algorithm accepts URL or document as an input. It extracts the sections like URL, meta, reference section, text before TOI, author and title using extractor layer. The output combined with classification layer output where it derives class labels of each document. The labels depict document type. These results are then passed on to processing layer for further cleaning and identification. Processing layer also creates trigram which comprises of POS sequence and later after feature identification for the published date it is annotated manually and trained for further usage. It shows learnability approach of our algorithm, but at the same time, it doesn't depend on the model for identifying published dates from day one and hence make it as an unsupervised framework. We start this

learnability approach once we collect 150000 unique trigrams for training. Fig2 shows these flows in details.

Table 1: Test Result Snapshot

URL	Our Algorithm	Alchemy	Result
http://www.alternrg.com/wp-content/uploads/2012/11/ALTER-NRG-NOV-2-2010.pdf	2010	Not Available	Correct
http://www.coronal.us/about-us/meet-the-team/	2000-01	Not Available	Correct
http://teesvalleyjobs.airproducts.co.uk/the-project/the-project.aspx	2015-10-12T18:14:02	Not Available	Correct
http://www.smsl.co.in/corpjourney.htm	2016-03-3T07:37:40	Not Available	Correct
http://www.ottawasun.com/2015/01/06/city-prefers-any-new-garbage-plant-to-be-energy-generator	2015	06-01-2015	Correct
http://www.hydrogenfuelnews.com/tag/fuel-cell-system/	2016	03-05-2016	Correct
http://waste2tricity.com/news/ervington-invests-in-waste2tricity.html	2012-11	2016(Wrong)	Correct
http://www.ottawasun.com/2014/12/13/time-to-get-serious-here-about-incineration	2014	13-12-2014	Correct
http://www.biodieselmagazine.com/blog/read/	2016	04-05-2016	Correct
http://www.afeservices.com/press_11152011_cgs.php	2005	Not Available	Correct
http://www.ecolateral.org/category/materials/food/	2015-05-5T06:40:48	Not Available	Correct
http://www.wastebusinessjournal.com/news/wbj20071127.htm	NA	NA	NA
http://www.ecolateral.org/category/geography/uk-regions/	2015-05-15T06:40:11	Wrong, 05-10-2009	Correct
http://www.peat.com/TVRC-introduction.html	2008	NA	Correct
http://www.google.co.in/patents/US6544530	2003-04-08	NA	Correct
https://www.google.com/patents/US7126032	2006-10-24	Wrong, 1931	Correct
http://www.google.co.in/patents/US8429103	2013-04-23	NA	Correct
http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3380258/	2012	04-2012	Correct
http://www.dailyenergyreport.com/tag/gasification/	2011-06-09	15-05-2016, Wrong	Correct
http://www.dailyenergyreport.com/tag/renewable/	2012	15-05-2016, Wrong	Correct
http://www.tudelft.nl/en/current/latest-news/article/detail/tu-delft-vindt-toilet-opnieuw-uit-voor-veilig-en-betaalbaar-sanitair/	2015-09-15	19-07-2011, Correct	Wrong
http://www.triplepundit.com/2014/03/caribbean-island-barbados-get-waste-energy-plant/	2014	26-03-2016	Correct
http://www.alternrg.com/wp-content/uploads/2013/08/Q2-2013-Results-Press-Release.pdf	2013	NA	Correct
http://www.environmental-expert.com/news/advanced-plasma-power-grants-industrial-licence-to-plasma-green-energy-292091	2012-05	2012-05-01	Correct
http://www.alternrg.com/wp-content/uploads/2012/11/ALTER-NRG-OCT-22.pdf	2012	NA	Correct
http://biofuelstp.eu/btl.html	2016-01-06	06-01-2016	Correct
http://www.alternrg.com/wp-content/uploads/2015/06/June-15-Offer-Extension.pdf	2015-06	NA	Correct
http://waste2tricity.com/company-and-people.html	2015-09-01T13:06:51	NA	Correct

Else {

V. COMPARISON

We compare our algorithm with Alchemy's. We tried to compare different type of URL ranging from Patent, News to Blogs. Table 2 shows the accuracy of our results and Table 1 shows few URL out of 6548 which we have tested. We have selected Alchemy's API for comparison as it widely used. We also found an interesting fact during our comparison with few other system is that mostly all system available for publication date only support HTML-based document as DOM manipulation performed on HTML. This is a key differentiator of our algorithm in comparison to others. We also performed testing on the corpus of memtracker[15]against 3864 URLS. Table 3 depicts the precision of our algorithm.

TABLE 2: ACCURACY MEASURE

#of URLS	Our Algo (Correct Result)	Alchemy (Correct Result)	Our Algo (Availability)	Alchemy (Availability)
6548	92.8%	80.5%	98%	75%

TABLE 3: PRECISION TABLE

# of URL	Precision	Recall	F-Measure (beta=1)
3864	0.82	0.71	0.76

VI. CONCLUSION

Our Algorithm shows the promising result in our test. We have illustrated published date extraction method in details in this paper. The other relevant dates can obtain in a similar manner with few changes in the rule engines.

REFERENCES

[1] Chen, Z., Ma, J., Rui, H., & Ren, Z. (2010, January). Web Page Publication Date Extraction and Application. Journal of Computational Information Systems.

[2] Prokhorenkova, L. O., Prokhorenkov, P., Samosvat, E., & Serdyukov, P. (2016). Publication Date Prediction through Reverse Engineering of the Web. WSDM 2016.

[3] Garcia-Fernandez, A., Ligozat, A.-L., Dinarelli, M., & Bernhard, D. (2011). When was it Written? Automatically Determining Publication Dates.

[4] Lopez, P. (2009). GROBID: Combining Automatic BibliographicData Recognition and Term Extraction ForScholarship Publications. Research and Advanced Technology for Digital Libraries, 13th European Conference, ECDL 2009. Corfu, Greece.

[5] Lopez, P. (2010). Automatic Extraction and Resolution of Bibliographical References in Patent Documents. Advances in Multidisciplinary Retrieval, First Information Retrieval Facility Conference, IRFC 2010. Vienna, Austria.

[6] Mario Lipinski, K. Y. (2013). Evaluation of Header Metadata Extraction Approaches and Tools for Scientific PDF Documents. (pp. 385-386). ACM/IEEE-CS.

[7] Gamon, M. (2004). Linguistic correlates of style: authorship classification with deep linguistic analysis features. COLING '04 Proceedings of the 20th International conference on Computational Linguistics. Stroudsburg, PA: Association for Computational Linguistics.

[8] Geva, R. (2015, December 13). Article's publication date extractor – an overview. Retrieved from blog.webhose.io: http://blog.webhose.io/2015/12/13/articles-publication-date-extractor-an-overview/

[9] Lu, Y., Meng, W., Zhang, W., & Yu, C. (2006). Automatic Extraction of Publication Time from News Search Results. Data Engineering Workshops, 2006. IEEE.

[10] Tannier, X. (2014). Extracting News Web Page Creation Time with DCTFinder. Irec-conf. Irec.

[11] Tkaczyk, D., Tarnawski, B., & Bolikowski, Ł. (2015, November). Structured Affiliations Extraction from Scientific Literature. D-Lib Magazine, pp. 11-12.

[12] Tekstosense. (2016). PDF Segmenter. Retrieved from TekstoSense: http://apis.tekstosense.com

[13] Wang, P., Domeniconi, C., & Hu, J. (2008, January). Cross-domain Text Classification using Wikipedia. IEEE Intelligent Information Bulletin.

[14] Bloom, N., Theune, M., & Jong, F. D. (n.d.). Using Wikipedia with associative networks for document classification. University of TWENTE. UTpublications.

[15] Raw MemeTracker phrase data. (n.d.). Retrieved from Memtracker: http://www.memetracker.org/data.html